



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification⁶:

A63F 9/22

A2

(11) International Publication Number:

WO 96/25989

(43) International Publication Date:

29 August 1996 (29.08.96)

(21) International Application Number: PCT/US96/02348

(22) International Filing Date: 23 February 1996 (23.02.96)

(30) Priority Data:

08/394,136

24 February 1995 (24.02.95)

US

(71) Applicant: VELOCITY, INC. [US/US]; Four Embarcadero #3100, San Francisco, CA 94111 (US).

(72) Inventors: MA, Moses; Apartment 21D, 1200 Gough Street, San Francisco, CA 94109 (US). KAPLAN, Stan; 3678 Vigilance, Rancho Palos Verdes, CA 90274 (US). WELDON, Robert, J.; 1256 29th, San Francisco, CA 94111 (US).

(74) Agents: STEINBERG, Robert et al.; Irell & Manella, Suite 900, 1800 Avenue of the Stars, Los Angeles, CA 90067 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

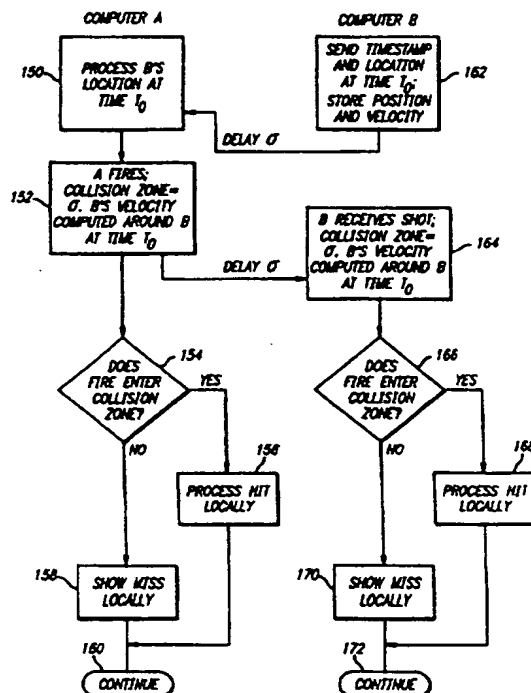
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD AND APPARATUS FOR MINIMIZING THE IMPACT OF NETWORK DELAYS

(57) Abstract

The methods and apparatus of the present invention determine expected network delays and alter program parameters based upon the determined delays. During the implementation of an interactive program, program parameters are adjusted based upon the expected delay and the program is not halted between transmission times. In one embodiment, with reference to computer games, the zone around which a game object is determined to be "hit" is expanded to account for the game object's motion during the delay period. In another embodiment, the effect of the network delay is minimized by not implementing a player's input on the player's computer until a specified time has elapsed.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

METHOD AND APPARATUS FOR MINIMIZING THE IMPACT OF NETWORK DELAYS

BACKGROUND OF THE INVENTION

5 1. Field of the Invention:

The present invention relates to interactive networks and, more particularly, to a method and apparatus for implementing computer programs, including games, over a network.

2. Art Background:

10 The popularity of interactive computer programs such as electronic games and the availability of computer networks has induced a demand for programs that allow network users to interact with one another. This demand will intensify as computer networks become more abundant and the number of users on those networks correspondingly increases. Network users may wish to interact with a central computer or with other users, using the network as a communications medium.

15 Certain types of communication networks introduce non-negligible delays in the transmission of information and program implementation over a network must account for these delays to synchronize the users' actions. For example, in the case of games, one player may have a space ship moving at a constant velocity and another player may shoot at that ship. The network delays the data containing the shot, and, by the time the computer playing
20 the game receives the data, the targeted space ship may have moved substantially from the position at which the shooting player last viewed the ship. Thus, the shooting player may believe that he has hit the spaceship but the other player's computer does not register a hit. This difference between the two players' perceptions will significantly derogate from their ability to enjoy the game. Network delays may similarly affect a variety of interactive
25 programs.

One method of synchronizing the users' actions is to halt program implementation during the transmission of data between a computer and a user. U.S. patent number 4,570,930 employs this method for games played on individual players' computers. Applying this method to the above example, the space ship would not move until the computer receives
30 the data transmitted by the shooting player. Halting game play, however, significantly decreases the speed of the game and distorts game play since the players will not perceive continuous action but will instead perceive frequently interrupted action.

The present invention overcomes the limitations of the prior art and provides a method and apparatus that allows interactive programs implemented over networks with finite delays to proceed quickly and with much less distortion than that introduced by any known prior art method.

5

SUMMARY OF THE INVENTION

The methods and apparatus of the present invention determine expected network delays and alter game parameters based upon the determined delays. To estimate the network delay at a given time for a particular terminal, signals are periodically sent from a central computer to the particular terminal. The delay for a particular signal is measured and a statistical analysis is performed on the delay times to determine an expected delay for a given time.

10

15

During the implementation of an interactive program, program parameters are adjusted based upon the expected delay and the program is not halted between transmission times. In one embodiment, with reference to computer games, the zone around which a game object is determined to be "hit" is expanded to account for the game object's motion during the delay period. In another embodiment, the effect of the network delay is minimized by not implementing a player's input on the player's computer until a specified time has elapsed.

20

In another aspect of the present invention, a central computer controls a game in which the winner may be predetermined by the central computer. The central computer sends a notification message to the winner without receiving any input from users. In this manner, a class of games may be played over a network with very long delays.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is an overview of a typical system for game play over a computer network.

FIGURE 2 is an example of a typical distribution of delays over a computer network.

5 **FIGURE 3** illustrates different expected delay distributions for different times of the day and different days of the week.

FIGURE 4 is a series of frames that illustrate the divergence in game play caused by network delays.

10 **FIGURE 5** is a series of frames that illustrate the implementation of "collision zone" expansion to eliminate the divergence caused by network delays.

FIGURE 6 is a flow chart of the present invention's delay method to eliminate the divergence caused by network delays.

FIGURE 7 is a flow chart of the present invention's "collision zone" expansion method to eliminate the divergence caused by network delays.

15

NOTATION AND NOMENCLATURE

The detailed descriptions which follow are presented largely in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art.

20

An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common
25 usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be

associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Further, the manipulations performed are often referred to in terms, such as adding or comparing, which are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein which form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar digital devices. In all cases there should be borne in mind the distinction between the method operations in operating a computer and the method of computation itself. The present invention relates to method steps for operating a computer in processing electrical or other (e.g., mechanical, chemical) physical signals to generate other desired physical signals.

The present invention also relates to apparatus for performing these operations. This apparatus may be specially constructed for the required purposes or it may comprise a general purpose computer as selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to a particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these machines will appear from the description given below.

DETAILED DESCRIPTION OF THE INVENTION

The present invention discloses methods and apparatus for implementing interactive computer programs over a network. Although the present invention is described with reference to specific block diagrams, and signals, etc., it will be appreciated by one of ordinary skill in the art that such details are disclosed simply to provide a more thorough understanding of the present invention. It will therefore be apparent to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well known circuits are shown in block diagram form in order not to obscure the present invention unnecessarily.

Figure 1 illustrates a typical system for interactive program implementation over a computer network that includes a central computer (also known as a server) 17 coupled to a plurality of individual users' terminals 8, 9 and 11 over a communications medium. The computer network comprises various nodes 10 - 26 and each user may communicate with the computer network through a different series of nodes. Thus, the delay between each terminal and the central computer may vary widely, depending upon various factors such as the number of nodes between the terminal and the central computer 17 and the volume of communication between those nodes.

Users may communicate over other types of well known network architectures. For example, the users' computers may communicate with each other over a peer to peer network. Alternatively, the users may communicate with each other over a network with distributed servers.

The individual users implement programs on their respective terminals 8, 9 and 11. The terminals receive data from the central computer 17, display the data on screen 28 and provide sound when appropriate. The terminals 8, 9 and 11 accept user input through a joystick 36, keyboard 32, mouse 34 or other input device and transmit user input to the central computer 17. As will be fully described below, in the present invention, the terminals 8, 9 and 11 do not merely display information provided by the central computer but must process program data.

The delay introduced by the network for data communication between the terminals 8, 9 and 11 and the central computer 17 presents a problem not inherent in interactive programs

in an environment where there is practically no delay humanly perceptible between the users' input to the computer and the computer's reception of that input and no delay between the computer's output and the reception of the output data by the display device. For example, assume that the network introduces a finite delay between the central computer 17 and the users' terminals 8, 9 and 11, and one user has a game object moving at a constant velocity. **Figure 4** illustrates the divergence in game play in a two player game, with players A and B, caused by the network delay.

As shown in block 60 of **Figure 4**, Player A has a ship 62 and player B has a ship 64. Blocks 60, 66, 68, 76, and 80 represent the display on player A's computer while blocks 70, 72, 74, 78 and 82 represent the display on player B's computer. At time t_0 , both players computers are synchronized, as shown in blocks 60 and 70, and it is assumed, without loss of generality, that both players' ships are stationary. After time t_0 , at time t_1 , player A fires a shot 84 at player B's ship as shown in block 66. At the same time, t_1 , player B moves his ship 64, as shown in block 72. Player B does not receive the shot information until after time t_2 , where $t_2 - t_1$ equals the network delay, as shown in block 78. Similarly, Player A does not receive the information regarding the movement of Player B's ship 64 until time t_3 , as shown in block 76. At time t_2 before time t_3 , however, player A perceives a "hit", as shown in block 68, while player B has not received the shot information at this time.

At time t_3 , player B's ship has been destroyed as perceived by player A, as shown in block 76. Conversely, at time t_3 , player B's terminal has just received the shot information, as shown in block 78. Subsequently, at time t_4 as shown in block 82, the position of the shot 84, which is identical to the position of the shot 84 as shown in block 68, is not identical with the position of Player B's ship 64, according to player B's terminal, as shown in block 82. At time t_4 , player A still perceives that player B's ship 64 has been destroyed. Thus, the network delay has caused a divergence in the results of game play. Network delays may cause many other types of divergences, apart from the example described with reference to **Figure 4**.

One prior art technique solved the divergence in interactive program implementation over networks by halting game play until the computer receives or should have received user input. This technique, however, significantly increases the amount of time required to implement a program. Further, this technique may distort the audio visual effect of the program, especially where the users have varying delays.

The methods of the present invention provides a substantial improvement over the prior art by not halting program implementation in between data transmission periods. In one embodiment, the method of the present invention allows the computers implementing a program to continuously run the program by adjusting aspects (eg. coordinates, sizes etc...) of program objects (e.g., spaceships, bombs etc...) to account for transmission delays from other users. The method comprises two fundamental steps: 1) computing the delays of each player; and 2) adjusting the aspects of program objects based upon the computed delays. The Specification will describe these steps separately. The Specification will then describe other methods for moderating the effect of network delays.

COMPUTING DELAYS

As shown in Figure 1, interactive program implementation may involve a number of users at various nodes on a network. The users communicate with the central computer through a number of intermediate nodes 10 - 26. Generally, users that must communicate through a larger number of intermediate nodes experience longer delays than those users that communicate through a relatively smaller number of intermediate nodes. The delay also depends upon the amount of data flow at the individual nodes, which may vary greatly between one intermediate node and another. The delay further depends upon the efficiency of the intermediate nodes' computers. Thus, each user probably will have a different delay that may vary over the course of the game and the central computer 17 must determine each user's delay.

Finally, delays are inherent in the transmission of information in the form of packets. A computer implementing a program over a network does not send a continuous bit stream to the other computers on the network. Instead, "routers" assemble information in packets and the routers send the packets. The process of assembling a packet introduces a delay in the transmission of data. On current X.25 or TCP/IP communication systems, this process produces a noticeable delay.

The method of the present invention generates statistics for each node in a system. As previously described, different nodes will experience different delays. Each node corresponds to a local phone call access number to the central computer 17 in the network. Before any user may log on to the network, the present invention utilizes the central computer 17 to determine the delays for each node and repeats this process to generate statistics for the delays of each node at different times of the day and different days of the week. Once the central computer 17 has sufficient statistics, as tested by the method of the present invention, users

may sign on to the network. The central computer 17 still samples delays even after users have signed on and updates each nodes' statistics accordingly.

According to one aspect of the present invention, the cost of generating the statistics is minimized. Whatever the communications medium between the central computer 17 and the node, there is a attendant cost to using that medium. Since a network may have many hundreds of nodes, it is important to minimize the amount of communication time between the central computer 17 and the nodes. Further, there is a cost associated with operating a computer and it is desirable to minimize this cost by minimizing the amount of computer time spent in generating proper statistics.

The method of the present invention generates sufficient statistics while minimizing the amount of computer and communication time spent generating the statistics. The time spent generating the statistics is directly proportional to the number of samples generated by the central computer 17. A statistical analysis is employed to minimize the number of samples while preserving sufficient information concerning the delays for each node.

If no users are logged on to the network, the central computer 17 generates delay samples by sending signals to each node. The nodes return the signal and the delay is proportional to the time elapsed between when the central computer 17 sent the signal and received the acknowledgement signal from a node. The delay from the central computer 17 to the node is assumed to be the same as the delay from the node to the central computer 17. Thus, the one way delay is measured by half the time elapsed between the sending and reception of signals by the central computer 17.

If users are logged on, the central computer 17 measures delays by examining the packets sent by the users. Each user's computer periodically sends packets to the central computer 17 with a time stamp that indicates when the packet was sent. The central computer 17 determines the delay as the difference between the time the central computer 17 received the packet and the time the packet was sent.

Turning now to the specific method for determining the proper number of samples, the expected delay distributions for a node will be described. As previously stated, since the delay a user experiences is a function of the traffic on a particular node and that traffic may vary even over a short time interval, a user will not experience a constant delay. Instead, over a short interval, such as five minutes, a user will experience a distribution of delays,

which will hereinafter be referred to as the actual delay distribution or ADD. The method of the present invention uses analytic forms to "model" or approximate the ADD. The following explanation will use a two parameter family of log normal distributions 44, as shown in Figure 2. It will be appreciated that the methods and apparatus of the present invention are applicable to all types of distributions not just log normal. The horizontal axis in Figure 2 represents the delay time and the vertical axis represents the probability density for each particular delay.

The log normal delay distributions may be expressed in parametric form by the equations:

$$d = \mu e^{\frac{z}{\sigma}}$$

and

$$P(d) = N(z),$$

where μ is the median, and σ is the "logarithmic standard deviation," $N(z)$ is the standard normal curve and $P(d)$ is the fraction of the area under the curve in Figure 2, below the value d . This distribution, so expressed, may also be written as $p(d) = \text{LN}(d, \mu, \sigma)$.

The distribution 44 as shown in Figure 2 will tend to change over time. A user will experience longer delays during those times where data flow over the network is high. For example, a user will probably experience longer delays on Monday at 5:00 p.m. than on Sunday at 11:00 p.m. Figure 3 illustrates possible distributions for a particular user at different times. The distributions 48 and 50 with longer delays than distribution 46 represent those times where network traffic is relatively high.

Before a player signs on, the method of the present invention generates a delay distribution, such as those shown in Figures 2 and 3, appropriate for that particular player at that particular time.

Before the method of the present invention is implemented, the central computer 17 has no information with regard to the distributions for each node. Upon implementation, the central computer 17 generates appropriate delay distributions for each node for each time interval within the period of a week. After one week has elapsed, each node has an

"expected" distribution for each interval in a week. Let μ' and σ' denote the parameters of the expected distribution.

5 The following algorithm updates each node's expected distribution for any time of any day and determines when sufficient statistics have been taken for a node before a player may sign on through that node. The following algorithm also generates the delay time that the central computer 17 will use to adjust game parameters.

10 As previously described, each node will have a distribution of delays, such as that shown in Figure 2, for each time interval. For example, between 4:55 p.m. and 5:05 p.m. on a Monday, each node will have its own distribution. The interval may be labelled by its mid-point, which in this example is 5:00 p.m. on Monday. The method of the present invention employs a five minute period before and after the specific time for a total sampling interval of 10 minutes. A person of ordinary skill in the art will recognize that other intervals may be chosen.

15 As previously described, a user will have an expected distribution for a particular time of a particular day.

20 The method of the present invention compares the user's expected distribution for a particular time with current samples of the delay for that time. If the current samples are sufficiently close to the expected distribution, it is considered that the expected distribution is confirmed and no more samples need be taken. Otherwise, more samples are taken until a new distribution, a modification of the expected distribution, has been adequately identified.

 The following is a mathematical treatment of the above algorithm. At each interval, each node has a typical distribution characterized by a median delay μ and a logarithmic standard deviation σ . A probability distribution for these two parameters will be established by sampling in each time interval.

25 Let $P_0(n,m|t)$ denote the prior probability that μ_n and σ_m are the correct values of the parameters at the time t . The distribution $P_0(n,m|t)$ is established by actual measurement of network delays. The posterior distribution $P(n,m|t-1)$ is obtained by updating the prior with the sampling data taken in interval t .

At the present interval, sampling has commenced at 5 minutes before the time t . A minimum number of P samples of the delay is taken. The particular set of values of samples is denoted as follows:

Let $E = \text{delay samples} = \{d, d_2, \dots, d_p\}$.

5 If the true parameters during this interval are μ and σ then the probability of obtaining evidence E is

$$\prod_{q=1}^P \frac{\sigma_m}{d_q} \exp\left\{-\frac{1}{2} \left(\sigma_m \ln \frac{d_q}{\mu_n}\right)^2\right\} = P(E | \langle \mu = \mu_n, \sigma = \sigma_m \rangle)$$

The posterior distribution for the present interval is then obtained as follows:

$$P_f(n, m | t) = \frac{P_o(n, m | t) P(E | \langle \mu = \mu_n, \sigma = \sigma_m \rangle)}{\sum_{m, n} P_o(n, m | t) P(E | \langle \mu = \mu_n, \sigma = \sigma_m \rangle)}$$

10

At this point, distribution $P_f(n, m | t)$ contains information concerning the current distribution based upon a node's history at t updated with information from current samples $\{d, d_2, \dots, d_p\}$.

15 To determine whether more samples for this interval must be taken, the distribution $P_f(n, m | t)$ is examined. As previously described, μ' and σ' denote the expected values of the parameters at this time. If P_f is sufficiently narrow around μ' and σ' , e.g. if $P_f(\mu', \sigma') \geq .90$, then the sampling is considered to have confirmed μ' and σ' as the correct distribution for this time, and no more sampling need be done. As will be readily appreciated by one of ordinary skill in the art, other threshold criteria may be chosen. If this threshold is not met, more
20 samples are taken until P_f converges to a (possibly new) μ, σ pair.

Next, from P_r the "updated" delay distribution for this time is calculated as a weighted function of the probability of achieving a particular distribution:

$$f(d|t) = \sum_{n,m} P_r(n,m|t) LN(d|\mu_n, \sigma_m)$$

The average delay for this node at this interval is:

$$\bar{d} = \int_{x=0}^{\infty} x f(x|t) dx$$

5 After a sufficient number of distributions have been generated for each node, the expected distributions are sufficiently stable to allow users to sign on. That is, the central computer 17 has sufficient statistics at each node to predict the delays for future intervals at that node. When a player signs on, the central computer 17 checks the delay distribution for that particular node at this time. If this distribution does not meet certain criteria, for example
10 if the average delay exceeds a certain threshold, the user will not be allowed to participate in an interactive program. Otherwise, during each interval, the computer implementing the program uses the delay distributions for that interval to control program parameters, as will be discussed below.

15 The delay distributions are continually updated after the users first sign on, using the method as previously described. This method of updating does not interfere with the program. Using the above algorithm ensures that the delay distributions are properly updated while minimizing computer and communications time without disturbing the flow of the program.

Using the Estimated Delay to Adjust Game Parameters

20 The following description assumes that the central computer 17 or the users' computers 8, 9 and 11 will use a particular delay, whether calculated from the delay algorithm described or derived by a different method. For example, the system may prompt the users for expected delays.

As previously described, delays introduced by computer networks interfere with the implementation of interactive programs over these networks. The method of the present invention adjusts program parameters to account for the network delay. In the following discussion, for ease of explanation, it is assumed that two players are playing a game over a network. The player that sends a fired object at the other player is hereafter defined as the "firing player" while the player whose game object is targeted is hereafter defined as the "receiving player." A game object may comprise any type of object that has a state that may be affected by an opposing player and a fired object may comprise any type of object, visible or invisible to the players, that may affect the state of an opposing player's game object.

A computer game with ships and fired shots illustrates an implementation of one aspect of the present invention. It will be appreciated that the present invention may be used in conjunction with any interactive program, including programs other than games, where the effect of users' input is time dependent. It will also be appreciated that the method of the present invention is applicable to networks with any number of users and over any type of network architecture.

As previously described, network delays cause the results of a game to diverge between the firing player and receiving player. This divergence occurs because the receiving player's computer continues to run the game while the data from the firing player is transmitted to the receiving player's computer. During this time, the receiving player's computer may have altered the receiving player's game object, for example a ship, from the position at which the firing player viewed the ship. In other words, the "state" of the game differs between the two participants. Thus, when the receiving player's computer receives the information concerning the location of the shot, the receiving player's ship may have substantially moved from the location of the shot. The firing player may have accurately fired the shot and thus the firing player's computer will register a hit. Conversely, the receiving player's computer will not register a hit, causing an apparent paradox.

The effects of this divergence may be eliminated by adjusting game parameters based upon the expected delay. A game parameter as used herein is any element of a game that may be adjusted to alter game play. More generally, a program parameter is any element of a program that may be adjusted to alter the results of the program. The extent of the alteration of program parameters depends upon the length of the delay. Shorter delays may require small adjustments while larger delays may require complicated adjustments.

If the typical delay interval is multiplied with the typical velocity of the targeted player's vehicle, the resulting quantity is defined as the "collision scale." If this collision scale is substantially smaller than the scale of the game object controlled by the user, then no correction is required. Any discrepancy will be hidden within acceptable limits, in terms of game play. However, should this collision scale be on the order or greater than the scale of the game object, additional corrections will be required.

"Collision" Zone Expansion

Should the collision scale be approximately of the same order as the game object, which for many systems could typically be on the order of two hundred milliseconds, the zone within which a "collision" is registered may be expanded. That is, to register that the receiving player's game object has been hit, instead of requiring an exact match between the current location of the receiving player's game object and the location of the fired object, the receiving player's computer may register a hit within a range of the location of the game object. Figure 5 illustrates one embodiment of "collision zone" expansion in a two player game with players A and B.

In Figure 5, blocks 100, 102, 104, 106 and 108 represent the display on player A's terminal while blocks 110, 112, 114, 116 and 118 represent the display on player B's terminal. At time t_0 , player A has a game object, which for purposes of illustration is shown as a ship 90, and player B has a ship 92 positioned on player A's terminal as shown in block 100. It is assumed that player B has moved his ship 92 prior to time t_0 but before the move information has been received by player A's terminal. Thus, as shown in block 110, player B's ship 92 at time t_0 as shown on player B's display is slightly different from the position on player A's terminal.

The position of B's ship at time t_0 is transmitted by player B's terminal and received by player A's terminal at time t_1 , where $t_1 - t_0$ constitutes the network delay. As shown in block 102, player A's terminal calculates an expanded "collision zone" in the form of a circle 94 around player B's ship 92 at the position of player B's ship 92 on player B's display at time t_0 which is now the position of player B's ship on player A's display at time t_1 . As shown in block 102, at time t_1 , player A fires a fired object which for purposes of illustration is shown as a shot and determines that player B's ship cannot escape the expanded collision zone 94. The shot data is transmitted to player B's terminal at time t_1 . At time t_1 , player B's terminal shows that player B's ship has moved from its position at time t_0 as shown in block 112.

At time t_2 , where $t_2 - t_1$ may be less than the network delay, player A's terminal displays the destruction 96 of player B's ship. At time t_2 , player B's terminal shows that player B's ship has moved from its position at time t_1 as shown in block 114. At time t_3 , where $t_3 - t_1$ is equal to the network delay, player B's terminal receives the shot information and computes an expanded collision zone 95 around player B's ship in its position 98 on player B's display at time t_0 , as shown in block 116. Player B's computer determines that the player A's shot falls within the expanded collision zone 94 and at time t_4 , player B's display shows the destruction 96 of player B's ship, as shown in block 118. In an alternative embodiment, the destruction of player B's ship as shown on player A's terminal, as shown in blocks 104 and 106, may be delayed until time t_4 such that both players simultaneously perceive the destruction of player B's ship.

The expanded collision zones 94 and 95 as shown in blocks 102 and 116 respectively of Figure 5 are a function of the network delay and the velocity of player B's ship. The radius of the circle 94 is expressed by the following formula:

$$\text{Radius} = \text{Speed} \times \text{Network Transfer Delay}$$

where speed is the speed of the relevant game object (e.g., player B's ship). The speed of the object multiplied by the network delay represents the maximum distance the object could have travelled between the time t_0 , when the position of the object is transmitted to the terminal of the firing player, and the time t_1 when the shot is fired. If a shot is fired within the collision zone radius, the range within which the receiving player's ship could have moved between t_0 and t_1 , then the shot fired within the zone will represent a hit. Otherwise, the shot will represent a miss.

Figure 7 is a flow chart of one possible implementation of "collision zone" expansion in a two player game with players A and B. Blocks 150, 152, 154, 156, 158 and 160 represent steps implemented on player A's computer while blocks 162, 164, 166, 168, 170 and 172 represent steps implemented on player B's computer. At block 162, player B's computer stores B's position and velocity at time t_0 and transmits this information to player A. Player A receives the information at block 150 and displays player B's location. At block 152, player A fires and player A's computer computes an expanded collision zone around player B at time t_0 where the collision zone radius = (B's Speed at time t_0) X Network Transfer Delay. The shot information is transmitted to player B's computer which computes the same collision zone at block 154. In an alternate embodiment, computer A may determine

at time t_1 , that the shot will hit or miss player B's ship and this information is provided to computer B. At blocks 154 and 166 respectively, the shot either enters or misses the collision zone. If the shot enters the collision zone, at blocks 156 and 168 respectively, computer A and computer B display a hit. If the shot fails to enter the collision zone, both computers
5 show a miss in blocks 158 and 170 respectively. The programs on each computer continue in blocks 160 and 172 respectively.

It will be appreciated that Figure 5 and the corresponding flow chart, Figure 7, illustrates one embodiment of collision zone expansion. Numerous alternatives are possible. For example, the collision zone may be expanded around player B's ship at a time other than
10 t_0 . As another example, a shot, rather than a ship, may be expanded based upon the network delay. Many other adaptations may be implemented in light of the foregoing description.

The expanded zone tends to cause more hits to register and the number of shots required to register complete victory may be increased to compensate for this effect. For example, if eight hits are required to destroy a game object absent a delay, 12 hits may be
15 required to destroy a game object if the collision zone is expanded. As will be appreciated by one of ordinary skill in the art, the proper increase in the number of shots required to destroy a target will depend upon various game parameters.

To make hits appear reasonable, the fired object may be expanded in size or otherwise modified. As previously described, a ship will be hit if the shot is within a certain range of
20 the ship. Thus, a ship may be hit by a shot that does not, to the receiving player, appear to actually hit the ship. To make the shot appear to hit the ship, the size of the shot may be expanded. Alternatively, the shot may appear to have an "electron cloud" zone around it and ships falling within the "cloud" will be hit. Further, the shot may break into fragments that scatter across a certain range when it is computed to be within the proximity of the target
25 ship. The trajectory of the fired object may be modified to hit or miss the game object. As will be appreciated by one of ordinary skill in the art, there are a variety of methods to expand the zone within which a ship appears to be hit. Alternatively, the speed of the vehicle or shots can be artificially reduced to compensate for delays. Thus the system ensures that the destruction of a ship appears reasonable to all of the players of the game.

30 For intermediate delays, between 300 and 600 milliseconds, further adjustments may be made. As previously described, the formula used for delays of approximately 200 to 300 milliseconds defines a circle within which a hit will be registered. For longer delays,

different geometrical shapes within which a hit may be registered may be employed to account for the velocity of the ship. If a ship is moving in a certain direction, it is more likely that ship will have been in a range along that direction. Thus, an ellipse rather than a circle may be used for intermediate delays for a ship with a velocity along the axis of the ellipse. If the players use a joystick, the damping control of the joystick, which increases the elapsed time between player action with the joystick and input from the joystick to the computer, may also affect the shape of the ellipse.

An ellipse will not adequately compensate for network delays longer than 600 milliseconds or for collision scales which are significantly greater than the scale of the vehicles, in excess of 600 milliseconds. For such longer delays or excessive collision scales, if the object is accelerating, the first order present velocity multiplied by the delay is not a good estimate of the vehicle's position. The longer delay allows the acceleration to become a significant determinant of the object's position. For longer delays, more information, such as the vehicle's velocity and acceleration, may be provided to the receiving computer to allow it to reconstruct the proper position of the object.

Game Parameters Other than Position

The methods and apparatus of the present invention may be employed to moderate or eliminate discrepancies not related to position caused by network delays. As previously described, position related discrepancies occur when terminals register different locations of the same program object at the same time. Other program parameters may diverge due to the presence of network delays.

Any parameter dependant upon elapsed time or past events may diverge due to network delays. For example, assume that two players, A and B, are playing a boxing simulation and the ability of a boxer to withstand a punch depends upon the "power" each boxer possesses. Assume further that the "power" of a boxer depends upon an input such as the touch of a button on a joystick. At time t_0 , player A throws a punch that knocks down player B, according to player A's terminal. Before player B receives the punch information, however, player B depresses a joystick button and increases the power of his boxer. When player B's terminal receives the punch data, player B has sufficient power to withstand the punch and her boxer is not knocked down. Thus, the players perceive different results.

Analogous to the methods related to position discrepancies, non-position related parameters may be adjusted according to a range the parameter could have had within the

delay period. In the boxing example previously discussed, when player B receives the "punch" data, player B's computer calculates the power level range player B's boxer could have had during the delay interval. If the power level required to knock down the boxer falls within this range, then player B's terminal displays player B's boxer as being knocked down.

- 5 More formally, if a game parameter changes as $d(\text{parameter})/dt$, then the derivative $d(\text{parameter})/dt$ evaluated at the time a terminal receives game data transmitted from a different terminal, multiplied by the delay time yields the change in the parameter value during the delay interval. Of course, certain game parameters may change impulsively so the derivative is not necessarily an approximation of an idealized derivative but may be calculated
- 10 statistically based upon the probabilities of a parameter change occurring in an interval and the effect of such a parameter change. Thus, if within a specified period T , a series of $n+1$ actions each has a probability p_i of occurring, and a given action is associated with a parameter change of x_i , then the expected parameter change for a delay d is:

$$(d/T) \sum_0^n p_i x_i$$

Game State Methods

- 15 Certain applications may require a more accurate correlation between the area within which the computer registers a "collision" and the location of the receiving player's game object. As previously described, expanding the collision zone tends to cause a divergence between the actual trajectory of a ship and the span of trajectories the ship could have taken that will cause the ship to be hit. In certain applications, it may be desirable not to increase
- 20 the zone within which a ship will be considered to have been hit. Alternatively, for these applications, the computer may store information concerning the past states of the system to recreate the sequence that would have occurred had the system not delayed data transfer.

Delayed Implementation Method

- 25 One possible method for recreating the game play that would have occurred without any network delay comprises the steps of extrapolating the trajectory of a target object to determine whether a fired object will hit the target object. In conjunction with extrapolation, the implementation of the sending player's control commands is delayed on the sending player's computer without stopping the game. In this case, when the sending player inputs a

control command to the computer, the sending player's computer does not implement the command until the receiving computer is anticipated to have received the control command. Instead, the sending player's computer determines, based upon the trajectory of the target object, whether the control command will result in a hit of the game object. The game control command is transmitted to the receiving computer. The receiving player's computer determines whether a hit will occur based upon the same trajectory information that the sending player's computer used to determine whether a hit will occur. The sending player's computer delays implementation of the control command to allow for the receiving player to receive the control command. Thus, the games remain synchronized, and any discrepancy can be eliminated by the two systems or a third controlling system.

Figure 6 is a flow chart of the delayed implementation method for a game played over a network. Blocks 110, 112, 114, 116, 118, 120 and 122 correspond to steps implemented on one computer, computer A, while blocks 124, 126, 128, 130 and 132 correspond to steps implemented on another computer, computer B. Without loss of generality, it is assumed that computer A's associated player, player A, fires a shot at a ship controlled by computer B's associated player, player B. At block 124, which corresponds to time t_1 , computer B sends to computer A a time stamp and the location of player B's ship at this time. At block 110, computer A receives the time stamp and location of player B's ship at time t_1 . In block 112, which corresponds to time t_2 , player A thereafter inputs a fire command and computer A sends to computer B data indicating that player A has fired at time t_2 .

Computer A does not implement the fire command at time t_2 . Instead, at a later time, t_3 , sound and animation that indicate that a shot has been initiated, but not the actual shot, are implemented on computer A. After a predetermined period, at time t_4 , computer A implements the traversal of the shot. The period $t_4 - t_3$ is selected to allow computer B to receive the shot command. The period $t_4 - t_3$ may be increased to model the time required to release a shot. For example, in real world applications, there is frequently a short delay between when a shot is initiated and when it actually begins its traversal such because the weaponry must process the shot information and prepare for the traversal of the shot. The period $t_4 - t_3$ may be selected to accurately model the delay of a real world weapon. As will be appreciated by one of ordinary skill in the art, the period $t_4 - t_3$ may model the delay of a real world device or be chosen for a fictional weapon.

5 The traversal initiated
 fire command input at time t_2 ;
 computer B determines whether
 position and velocity of player
 player B after time t_1 are ignored
 position and velocity at time t_1
 hit of player B's ship. Alternately
 upon its position and velocity is
 displayed.

10 Since player B may have
 B's ship may not equal the position
 computer registers a hit where
 sending player's computer differs
 t_4 , a strategy must be invoked in
 computer such that the receiving
 perceives a hit when the missile
 ship at the same time. Similarly
 should generate, and the player

20 To adjust what occurs is
 calculated by the receiving player
 define the trajectory actually displayed
 player's computer calculates a
 the trajectory of the missile that
 appears at time t_4 . Similarly, if
 receiving player would perceive
 to a different location at time t_4
 appreciated by a person of ordinary
 trajectory of missiles to either

30 Computer A also displays
 time t_1 . In block 118, computer
 based upon its position and velocity
 branches to block 122 and continues
 shot would not have hit player

128 branches to block 132 and a miss is displayed. The shot may be adjusted on player A's display to make a hit or miss appear reasonable. Since both computers determine whether a hit would have occurred based upon the same time, position, velocity data, the computers obtain identical results and both display either a hit or a miss.

5 Although the sending players' computer, computer A in the above example, does not implement the control signal, for example a fire command, until after the expected delay has elapsed, the computer does not stop the game during this period. Thus, the sending player perceives continuous game action except that there is a delay between the time the sending player inputs a control action and the time the results of that action appear on the sending
10 player's display screen. If the expected delay is relatively small, the delay between the time that the sending player inputs the command to the computer and the time the computer implements the command on the display screen will not be perceived by the sending player.

 Another strategy for synchronizing the timing of the hit or miss on the computers is to decrease the speed of the fired object on player A's display relative to the speed of the fired
15 object on player B's display. Alternatively, the trajectory of the fired object on player A's display may be modified to increase the amount of time required to hit or miss the fired object. In either case, the speed or trajectory should be adjusted such that the total traversal time on player A's computer equals the traversal time on player B's computer plus the network delay.

20 As will be appreciated by one of ordinary skill in the art, the delayed implementation method is not limited to the determination of whether a ship has been hit by a fire command. The delayed implementation method may be employed to correct for a network delay in any interactive network program where the effect of user input is time dependent.

Pseudo-Interactive Games

25 For extremely long delays, or collision zones far in excess of the scale of the virtual vehicles, other, more drastic measures must be taken in order to insure apparently responsive game play. Certain games may be played by eliminating interactivity, but providing a sense of interactivity where a central computer controls game play.

 For example, delays of over several minutes would render traditional Bingo
30 unplayable. According to one aspect of the present invention, the central computer stores all of the information required to determine a winner. For example, for Bingo, the central

computer stores information concerning each players' card. The computer checks each card after it generates a new bingo slot to determine if a player has won. If so, the computer sends a message to that the winning player or pre-allocates back channel in an unresponsive network system that does not provide immediate messaging bandwidth to insure that the winning player can announce her victory. Except in the case of more than one potential winning player, the computer never waits for input from the players' terminals, which greatly minimizes problems caused by network delays.

When there is more than one potential winning player, the winner is determined based upon response time. The potential winning players are notified that they have potentially won. This notification usually will not occur simultaneously since the delays from the central computer to the players' terminals are distinct. The potential winning players respond and the central computer receives the responses of the potential winning players. From the time the central computer sent out the notification that a player has potentially won, the central computer receives the responses after the delay from the central computer to a potential winning player in addition to the player's response time and in addition to the delay from the potential winning player to the central computer. Thus, to calculate the response time, the central computer subtracts the known delays to and from a player's terminal from the total time from transmission of the notification to the receipt of a players' response. The player with the shortest response time is then deemed the actual winner.

This method is especially useful for those networks with very long back channel delays (from the players to the central computer) but relatively small forward delays.

While the invention has been described in conjunction with the preferred embodiment, it is evident that numerous alternatives, modifications, variations and uses will be apparent to those skilled in the art in light of the foregoing description. For example, the present invention may be implemented over an interactive television system where a television screen comprises the display screen for the interactive program. Many other adaptations of the present invention are possible.

Claims:

1. A method for minimizing the distortion in interactive programs over a computer network caused by transmission delays over the network, the interactive program involving input from at least two computers that communicate with one another over the network, the method including the steps of:
 - 5 transmitting command data from a first computer to a second computer;
 - adjusting a program parameter on the second computer to compensate for an estimated transmission delay; and
 - implementing the command on the second computer.
2. The method of claim 1 wherein the estimated transmission delay is the transmission delay between the first computer and the second computer.
3. The method of claim 1 wherein the estimated transmission delay is a multiple of the transmission delay between the first computer and the second computer.
4. The method of claim 1 wherein the program comprises a game in which a first game object may interact with a second game object within a collision zone.
5. The method of claim 4 wherein the step of adjusting a program parameter on the second computer further includes the step of altering the collision zone.
6. The method of claim 5 wherein the first game object comprises a target object and the second game object comprises a fired object capable of hitting the target object.
7. The method of claim 6 wherein the target objects may move.
8. The method of claim 7 wherein the step of altering the collision zone further includes the step of expanding the area around a target object within which the second computer will register a hit of the target object.
9. The method of claim 8 wherein the expanded area around the target object is a circle, the radius of the circle defined by the product of the estimated transmission delay and the speed of the target object.

10. The method of claim 9 further including the step of transmitting to the first computer from the second computer data including information whether the second computer registered a hit.

11. The method of claim 9 further including the step expanding the collision zone around the target object within which the first computer will register a hit of the target object.

12. The method of claim 11 wherein the collision zone is expanded around the target object on the first and second computers based upon the target object's position at the same time.

13. The method of claim 9 wherein the fired object is modified on the second computer.

14. The method of claim 13 wherein the fired object is modified in the form of an "electron cloud."

15. The method of claim 8 wherein the expanded area around the target object is an ellipse.

16. The method of claim 1 wherein the step of adjusting a program parameter on the second computer includes adjusting a program parameter that is a function of time based upon a multiple of the estimated transmission delay and the derivative of the program parameter with respect to time.

17. A method for minimizing the distortion in interactive programs over a computer network caused by transmission delays over the network, the interactive program involving input from at least two computers that communicate with one another over the network, the method including the steps of:

- 5 transmitting command data from a first computer to a second computer at a time t_0 ;
receiving the command data on the second computer at a time t_1 ;
on the second computer, receiving response data at a time t_2 ;
on the second computer, determining the result of the response data and the command
data;
10 transmitting the determined result to the first computer; and
implementing the determined result on the first computer and the second computer.

18. The method of claim 17 wherein the program is a game, the command data is position data of a game object and the response data is position data of a fired object.

19. The method of claim 18 wherein the step of determining the result of the response data and the command data further includes the step of determining whether the fired object will collide with the target object based upon the trajectory of the target object.

20. The method of claim 19 further including the step of adjusting the trajectory of the fired object to collide with a current position of the target object on the first computer.

21. A method for minimizing the distortion in interactive programs over a computer network caused by transmission delays over the network, the interactive program involving input from at least two computers that communicate with one another over the network, the method including the steps of:

5 receiving program control information on a first computer;
 transmitting the program control information from the first computer to a second computer;
 continuing at least one aspect of the program on the first computer before the second computer receives the program control information transmitted from the first computer;
10 delaying implementation of the program control information on the first computer; and
 implementing the program control information on the first computer after a predetermined period.

22. The method of claim 21 wherein the program comprises a game in which a first game object may interact with a second game object within a collision zone and the program control information comprises game control information.

23. The method of claim 22 wherein the game control information comprises the shot of a fired object.

24. The method of claim 23 wherein the step of continuing at least one aspect of the program on the first computer includes implementing sound and animation of the shot on the first computer.

25. The method of claim 24 wherein the step of implementing the program control information on the second computer after a predetermined period includes the step of initiating the traversal of the fired object.

26. The method of claim 23 further including the steps of:
on the first computer, determining whether the fired object collides with the target object based upon the target object's position at a first time; and
on the second computer, determining whether the fired object collides with the target object based upon the target object's position at a second time.

27. The method of claim 26 wherein the first time is equal to the second time.

28. The method of claim 26 wherein the first time and the second time are before the time the program control information is received on the first computer.

29. A method for determining a winning player in a multi-player game played over a computer network, the computer network including a central computer and a plurality of terminals in communication with the central computer wherein there is a delay for signals that are communicated between the central computer and the terminals, the method including the steps of:

sending game data from the central computer to the plurality of terminals;
predetermining a winning player; and
sending a message from the central computer to the terminal of the winning player after the game is complete.

30. The method of claim 29 wherein the game is bingo.

31. A method for determining rank of responses from users in a multi-user communication system, the system having a central computer and a plurality of terminals in communication with the central computer wherein there is an output delay for signals that are communicated from the central computer and the plurality of terminals and an input delay for signals that are communicated from the plurality of terminals to the central computer, the method including the steps of:

sending a first signal to at least two of the plurality of terminals;
receiving responses from at least two of the plurality of terminals;

- for each of at least two received responses, determining a rank for the received
10 response by:
- determining a total time defined by the time the first signal was sent
and the time the received response was received;
 - 15 determining a delay time defined by the addition of the output delay to
the terminal sending the response and the input delay from the terminal
sending the response;
 - determining a response time by subtracting the delay time from the
total time; and
 - ranking the response according to the response time.

32. An apparatus for minimizing the distortion in interactive programs over a computer
network caused by transmission delays over the network, including:
- a first computer;
 - a second computer including means for adjusting a program parameter to compensate
5 for an estimated transmission delay; and
 - a computer network for transmitting between the first computer and the second
computer.
33. The apparatus of claim 32 wherein the estimated transmission delay is the transmission
delay between the first computer and the second computer.
34. The apparatus of claim 32 wherein the estimated transmission delay is a multiple of
the transmission delay between the first computer and the second computer.
35. The apparatus of claim 32 wherein the first and second computers include means for
implementing a program that comprises a game in which a first game object may interact with
a second game object within a collision zone.
36. The apparatus of claim 35 wherein the means for adjusting a program parameter on
the second computer further includes means for of altering the collision zone.
37. The apparatus of claim 36 wherein the first game object comprises a target object and
the second game object comprises a fired object capable of hitting the target object.
38. The apparatus of claim 37 wherein the target objects may move.

39. The apparatus of claim 38 wherein the means for altering the collision zone further includes means for expanding the area around a target object within which the second computer will register a hit of the target object.
40. The apparatus of claim 39 wherein the means for expanding the area around a target object include means for generating a circle, the radius of the circle defined by the product of the estimated transmission delay and the speed of the target object.
41. The apparatus of claim 39 wherein the first computer includes means for expanding the collision zone around the target object within which the first computer will register a hit of the target object.
42. The apparatus of claim 41 wherein the means for expanding the collision zone on the first and second computers further include means for expanding the collision zone around the target object on the first and second computers based upon the target object's position at the same time.
43. The apparatus of claim 39 wherein the second computer further includes means for modifying the fired object.
44. The apparatus of claim 43 wherein the means for modifying the fired object further include means for modifying the fired object in the form of an "electron cloud."
45. The apparatus of claim 39 wherein the means for expanding the area around the target object further include means for expanding the area in the form of an ellipse.
46. The apparatus of claim 32 wherein the means for adjusting a program parameter on the second computer includes means for adjusting a parameter that is a function of time based upon a multiple of the estimated transmission delay and the derivative of the parameter with respect to time.
47. An apparatus for minimizing the distortion in interactive programs over a computer network caused by transmission delays over the network, including:
a first computer including:
means for sending command data;

5

means for implementing results determined by

a different computer;

a second computer including:

means for receiving command data sent by the first computer;

means for receiving response data input by a user;

10

means for determining the result of the response data and the command data;

means for sending the determined result to the first computer;

means for implementing the determined result; and

a computer network for transmitting data between the first computer and the second computer.

15

48. The apparatus of claim 47 wherein the program is a game, the command data is position data of a game object and the response data is position data of a fired object.

49. The apparatus of claim 48 wherein the means for determining the result of the response data and the command data further includes means for determining whether the fired object will collide with the target object based upon the trajectory of the target object.

50. The apparatus of claim 49 wherein the first computer further includes means for adjusting the trajectory of the fired object to collide with a current position of the target object on the first computer.

51. An apparatus for minimizing the distortion in interactive programs over a computer network caused by transmission delays over the network, including:

a first computer;

a second computer including:

5

means for receiving program control information;

means for sending the program control information to the first computer;

means for continuing at least one aspect of the program on the first computer before the first computer receives the program control information;

10

means for delaying implementation of the program control information; and

means for implementing the program control information after a predetermined period; and

15 a computer network for transmitting data between the first computer and the second computer.

52. The apparatus of claim 51 wherein the program comprises a game in which a first game object may interact with a second game object within a collision zone and the program control information comprises game control information.

53. The apparatus of claim 52 wherein the game control information comprises the shot of a fired object.

54. The apparatus of claim 53 wherein the means for continuing at least one aspect of the program includes means for implementing sound and animation of the shot.

55. The apparatus of claim 53 wherein the means for implementing the program control information after a predetermined period further includes means for initiating the traversal of the fired object.

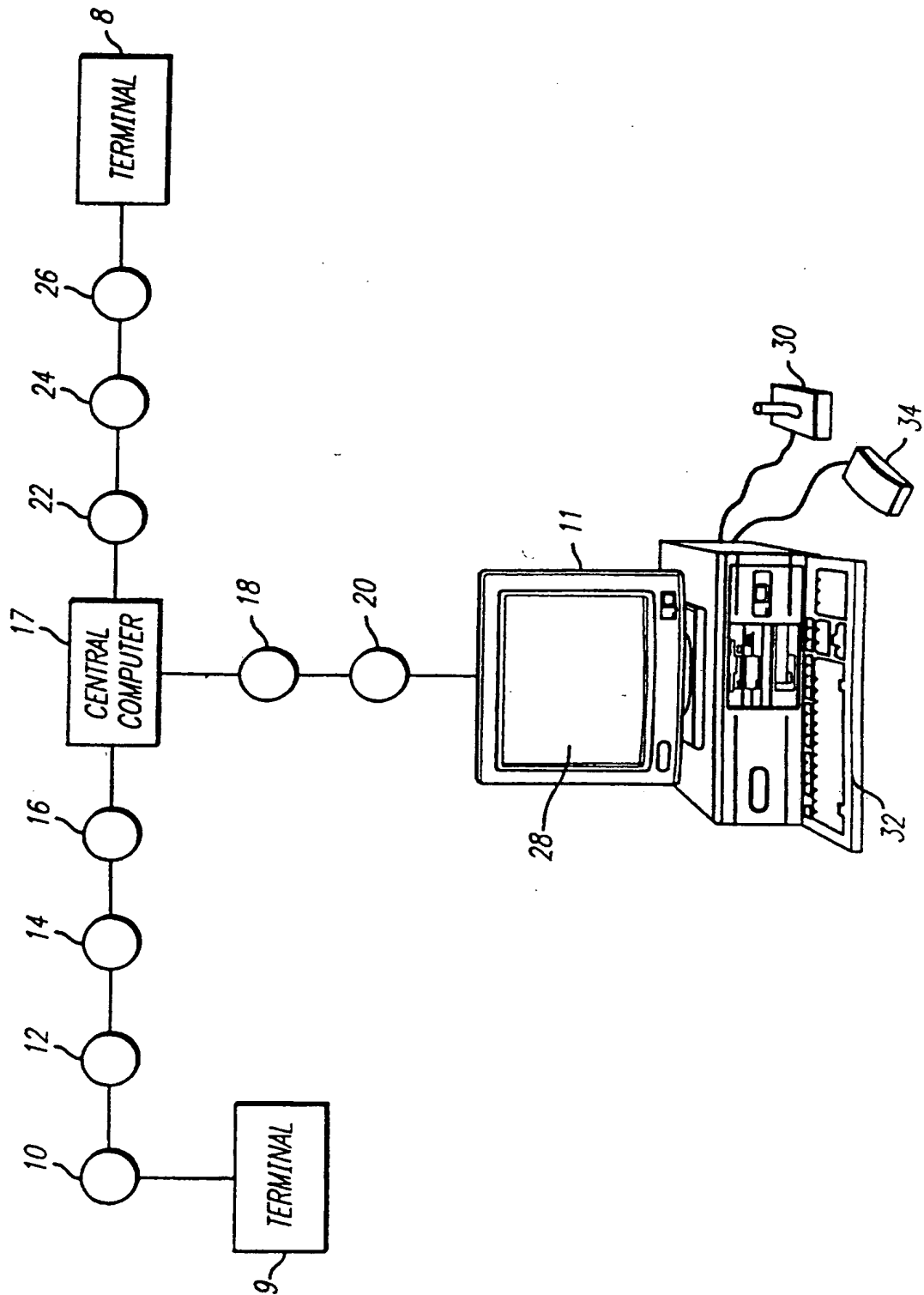
56. The apparatus of claim 53 wherein:

the first computer further includes means for determining whether the fired object collides with the target object based upon the target object's position at a first time; and

5 the second computer further includes means for determining whether the fired object collides with the target object based upon the target object's position at a second time.

57. The apparatus of claim 56 wherein the first time is equal to the second time.

FIG. 1



2/6

FIG. 2

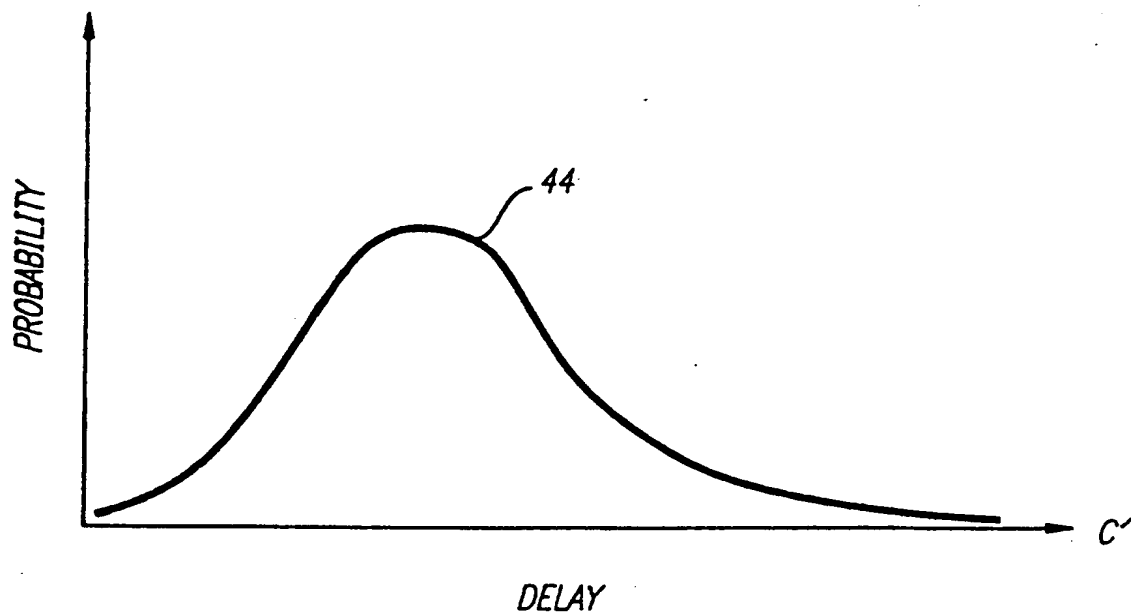


FIG. 3

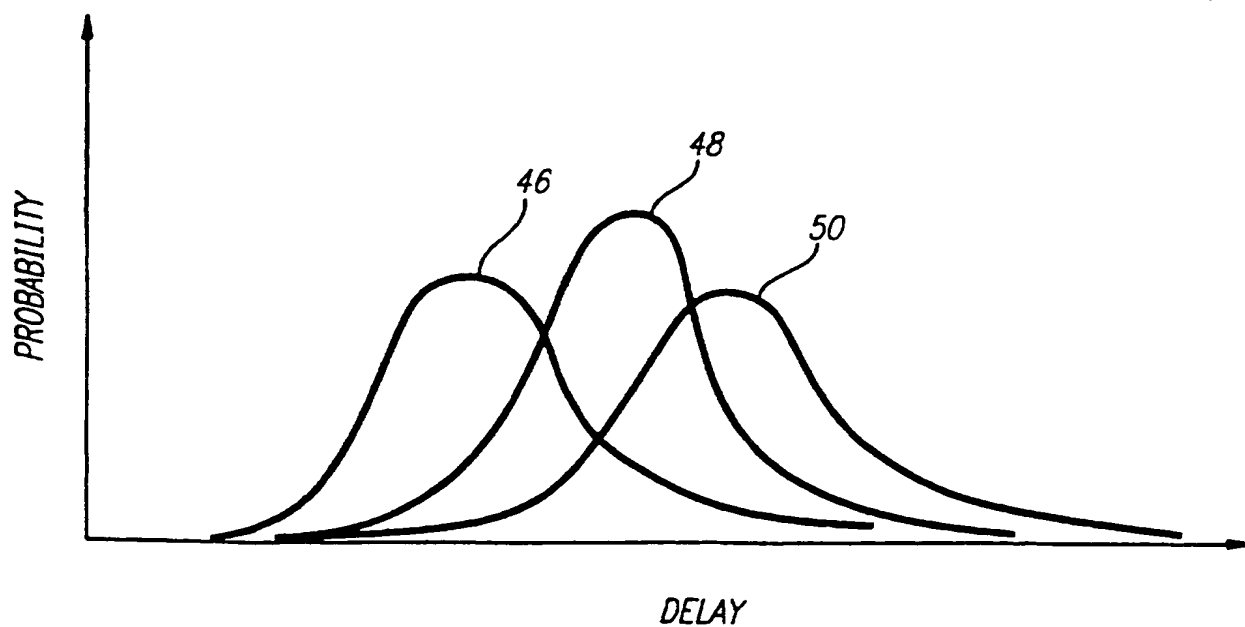


FIG. 4

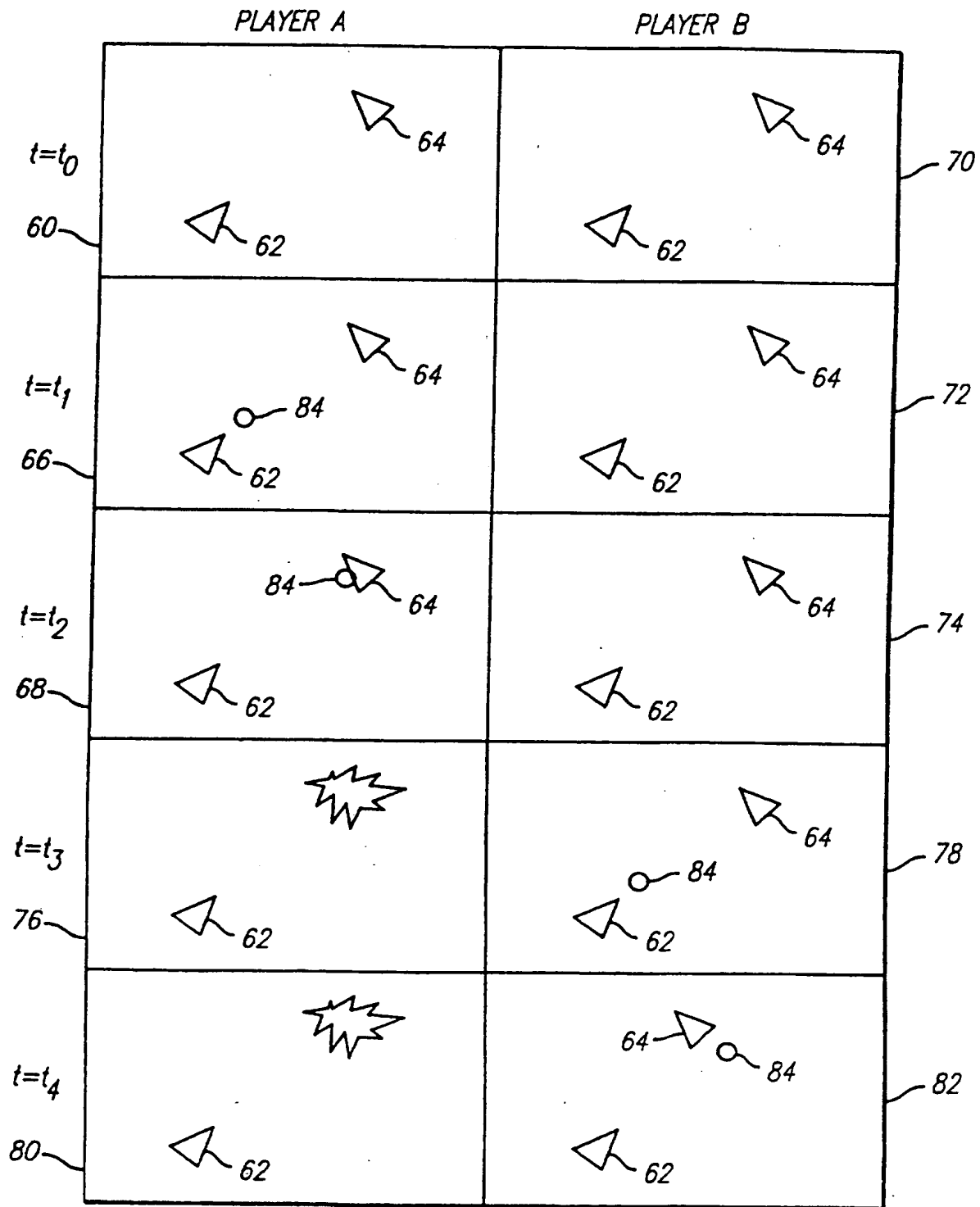
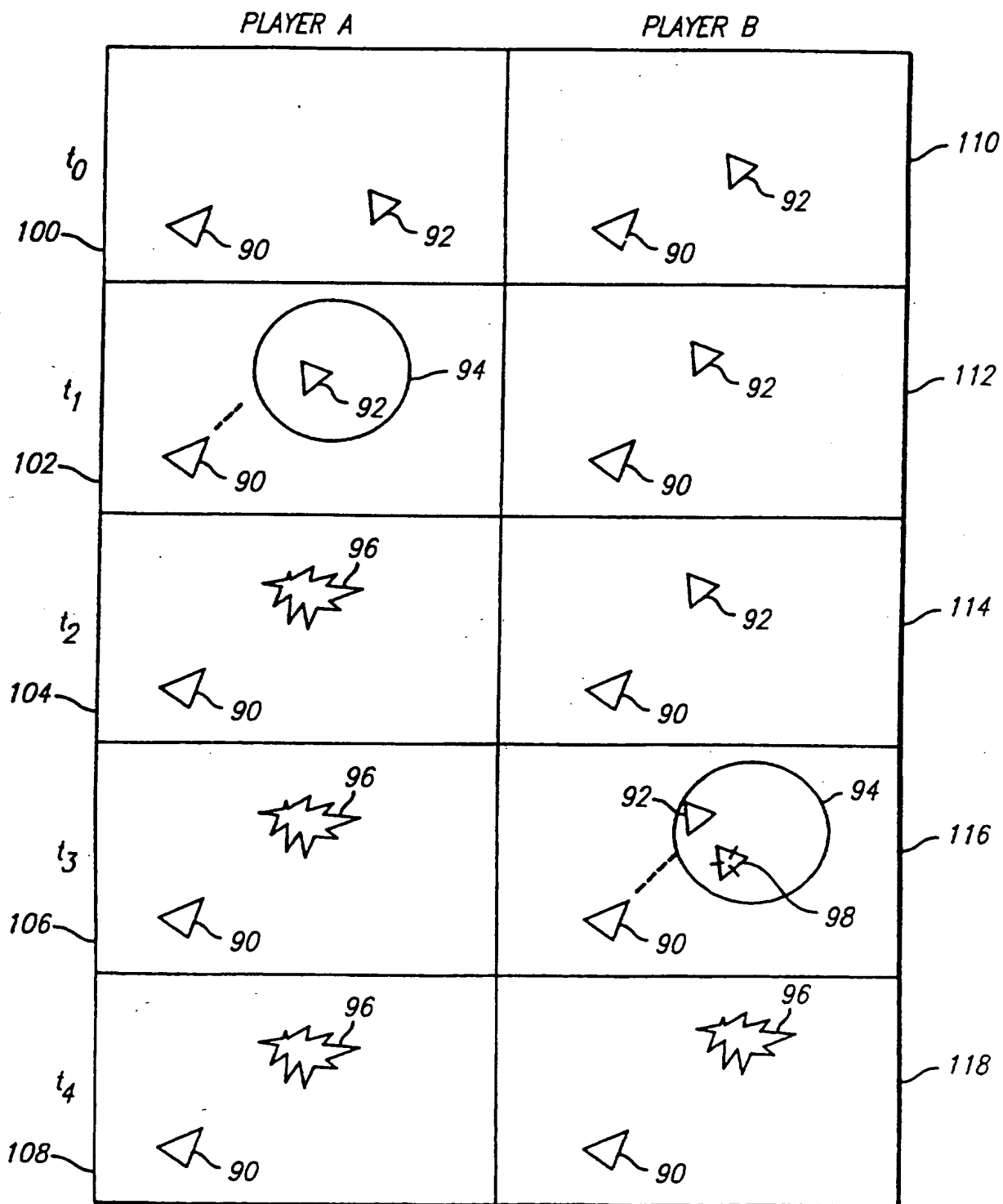


FIG. 5



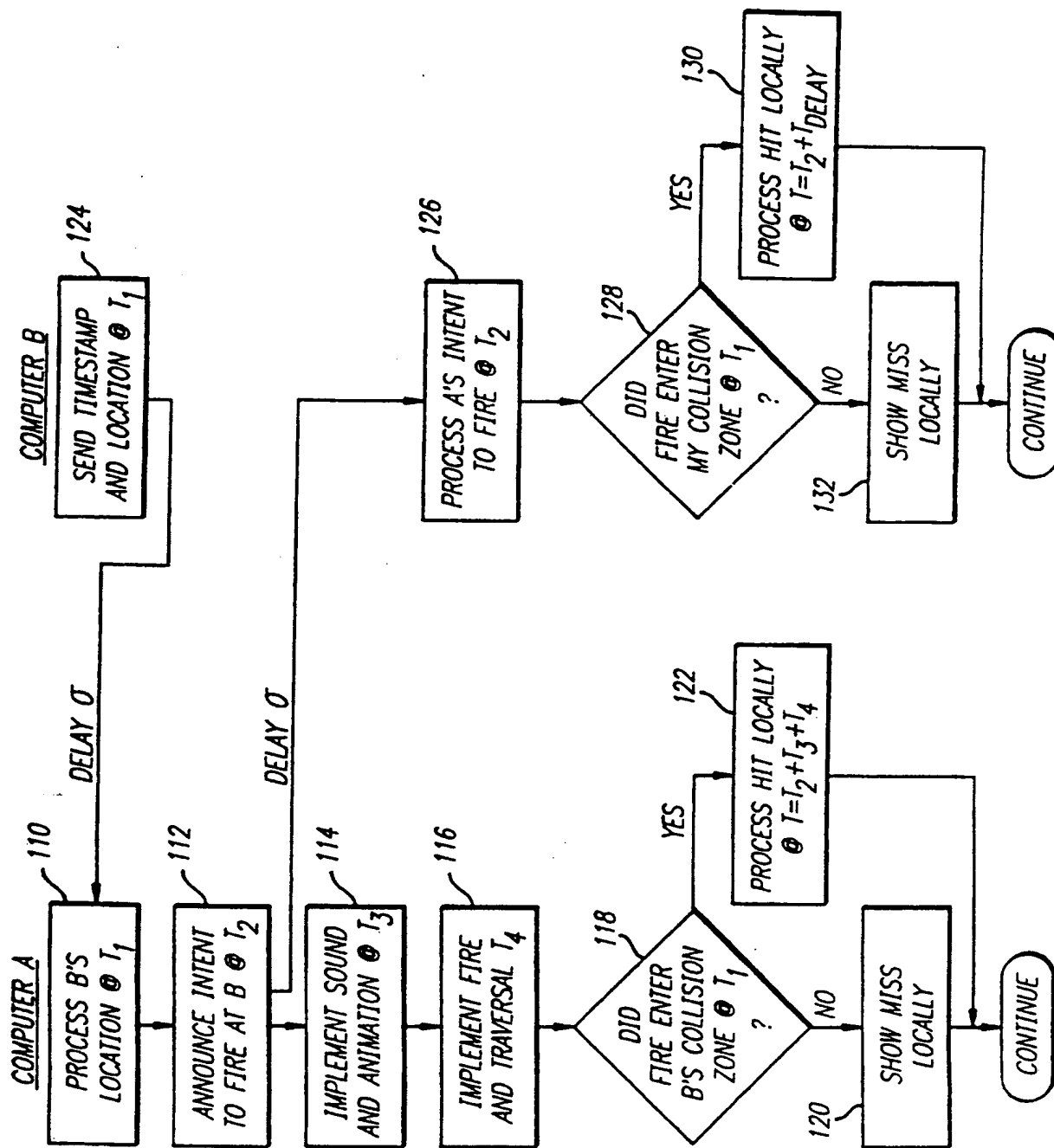


FIG. 6

6/6

FIG. 7

